

**RECEIVED  
CENTRAL FAX CENTER****APR 02 2007**Attorney Docket No.: 40101/13901  
Ref. No.: 2008.005**REMARKS****I. INTRODUCTION**

Claims 1-7 and 10-68 are pending in the present application. No new matter has been added. In view of the following remarks, it is respectfully submitted that all of the pending claims are in condition for allowance.

**II. OBJECTION TO THE SPECIFICATION**

The Specification stands objected to for failing to properly note continuity data. (*See* 11/3/06 Office Action, p. 2.) Applicants note that the application cited by the Examiner, PCT/US04/31371, is an application that claims priority to the present application rather than the reverse. Applicants respectfully submit that they are not required to reference, in the Specification, an application that claims priority to the present application. (*See* 37 C.F.R. § 1.57.) Therefore, no amendment to the Specification is required and the objection to the Specification should be withdrawn.

**III. CLAIM REJECTIONS – 35 U.S.C. § 102(a)**

Claims 1-7 and 10-68 stand rejected under 35 U.S.C. § 102(b) as anticipated by U.S. Patent 5,136,709 to Shirakabe et al. (“Shirakabe”). (*See* 11/3/06 Office Action, pp. 4-14.)

Claim 1 recites “[a] system for dynamically linking application code created by a programmer into a running operating system kernel, comprising: an environment library comprising one or more routines for insulating the application code from the operating system environment and for implementing a uniform execution environment; and a build system for *constructing a loadable module from the application code and the environment library and for*



Attorney Docket No.: 40101/13901  
Ref. No.: 2008.005

*constructing a standard executable program from the loadable module and an execution library, wherein the execution library comprises one or more routines for transparently loading the loadable module into the running operating system kernel, passing arguments to the loadable module, and terminating and unloading the loadable module after receiving a termination signal."*

Shirakabe describes a method for generating an operating system. A driver 5, a linkage library 12, and a driver definition table 14 are combined by the use of a static linkage editor 7. (See Shirakabe, col. 5, ll. 7-11; Fig. 1.) The output of this combination is load module 26. (See *id.*) The Examiner contends that the process performed by the static linkage editor 7 and illustrated in Fig. 1 of Shirakabe anticipates the build system of claim 1. (See 11/3/06 Office Action, p. 4, ll. 20-27.) However, the static linkage editor 7 is different from the build system of claim 1.

First, once the build system of claim 1 has constructed the loadable module, it continues by "constructing a standard executable program from the loadable module and an execution library." Conversely, Shirakabe does not disclose performing further modifications to the load module 26; rather, it is used as a finished product. (See Shirakabe, cols. 5-7.) Further, Shirakabe does not state that the load module 26 is executable, nor does it discuss the creation of an executable program from the load module 26.

Therefore, it is respectfully submitted that Shirakabe does not disclose "a build system for constructing a loadable module from the application code and the environment library and for constructing a standard executable program from the loadable module and an execution library," as recited in claim 1. Accordingly, the rejection of claim 1 should be withdrawn. Because claims 2-4 and 10-20 depend from, and, therefore, include all of the limitations of claim 1, it is respectfully submitted that these claims are also allowable for at least the reasons stated above.



Attorney Docket No.: 40101/13901  
Ref. No.: 2008.005

Claim 5 recites “[a] method, comprising: creating a loadable module; creating an executable program; and executing the executable program, wherein *the executable program performs a method comprising the steps of: setting up input/output channels; inserting the loadable module into an operating system address space, wherein, once the loadable module is inserted into the operating system address space, the loadable module begins to execute; and waiting for the loadable module to connect via kernel/user channels and then connecting those kernel/user channels to the input/output channels.*”

For the reasons previously discussed with reference to claim 1, it is respectfully submitted that Shirakabe does not disclose “the executable program performs a method comprising the steps of: setting up input/output channels; inserting the loadable module into an operating system address space, wherein, once the loadable module is inserted into the operating system address space, the loadable module begins to execute; and waiting for the loadable module to connect via kernel/user channels and then connecting those kernel/user channels to the input/output channels,” as recited in claim 5. Accordingly, the rejection of claim 5 should be withdrawn. Because claims 6 and 7 depend from, and, therefore, include all of the limitations of claim 5, it is respectfully submitted that these claims are also allowable for at least the reasons stated above.

Claim 21 recites “[a] computer readable medium having computer instructions stored thereon, the computer instructions comprising: a first set of computer instructions for insulating application code from an operating system environment; a second set of computer instructions for constructing a loadable module from the application code and the first set of computer instructions; and *a third set of computer instructions for constructing an executable program from the loadable module and a fourth set of computer instructions, wherein the fourth set of computer instructions includes computer instructions for transparently loading the loadable module into a running operating system kernel, passing arguments to the loadable module, and*



Attorney Docket No.: 40101/13901

Ref. No.: 2008.005

*terminating and unlading the loadable module from the running operating system kernel after receiving a termination signal."*

For the reasons previously discussed with reference to claim 1, it is respectfully submitted that Shirakabe does not disclose "a third set of computer instructions for constructing an executable program from the loadable module and a fourth set of computer instructions, wherein the fourth set of computer instructions includes computer instructions for transparently loading the loadable module into a running operating system kernel, passing arguments to the loadable module, and terminating and unlading the loadable module from the running operating system kernel after receiving a termination signal," as recited in claim 21. Accordingly, the rejection of claim 21 should be withdrawn. Because claims 22-33 depend from, and, therefore, include all of the limitations of claim 21, it is respectfully submitted that these claims are also allowable for at least the reasons stated above.

Claim 34 recites "[a] computer system, comprising: first means for insulating application code from an operating system environment; second means for constructing a loadable module from the application code and the first means; *third means for constructing an executable program from the loadable module*; and fourth means for transparently loading the loadable module into a running operation system kernel, passing arguments to the loadable module, and terminating and unloading the loadable module from the running operating system kernel after receiving a termination signal."

For the reasons previously discussed with reference to claim 1, it is respectfully submitted that Shirakabe does not disclose "third means for constructing an executable program from the loadable module," as recited in claim 34. Accordingly, the rejection of claim 34 should be withdrawn. Because claims 35-46 depend from, and, therefore, include all of the limitations of claim 34, it is respectfully submitted that these claims are also allowable for at least the reasons stated above.



Attorney Docket No.: 40101/13901  
Ref. No.: 2008.005

Claim 47 recites “[a] computer system for dynamically linking application code created by a programmer into a running operating system kernel, comprising: means for creating a loadable module; and *means for creating an executable program that is configured to perform a method comprising the steps of: setting up input/output channels; inserting the loadable module into address space of the running operating system kernel, wherein, once the loadable module is inserted into the address space, the loadable module begins to execute; and waiting for the loadable module to connect via kernel/user channels and then connecting those kernel/user channels to the input/output channels.*”

For the reasons previously discussed with reference to claim 1, it is respectfully submitted that Shirakabe does not disclose “means for creating an executable program that is configured to perform a method comprising the steps of: setting up input/output channels; inserting the loadable module into address space of the running operating system kernel, wherein, once the loadable module is inserted into the address space, the loadable module begins to execute; and waiting for the loadable module to connect via kernel/user channels and then connecting those kernel/user channels to the input/output channels,” as recited in claim 47. Accordingly, the rejection of claim 47 should be withdrawn. Because claims 48-51 depend from, and, therefore, include all of the limitations of claim 47, it is respectfully submitted that these claims are also allowable for at least the reasons stated above.

Claim 52 recites “[a] method for dynamically linking application code created by a user into a running operating system kernel, comprising: constructing a loadable module from application source code written by a user; *creating an executable program, wherein the executable program is configured to transparently load the loadable module into the running operating system kernel*; executing the executable program, thereby loading the loadable module into the running operating system kernel; and unloading the loadable module from the running operating system kernel by sending a termination signal to the executable program.”



Attorney Docket No.: 40101/13901  
Ref. No.: 2008.005

For the reasons previously discussed with reference to claim 1, it is respectfully submitted that Shirakabe does not disclose "creating an executable program, wherein the executable program is configured to transparently load the loadable module into the running operating system kernel," as recited in claim 52. Accordingly, the rejection of claim 52 should be withdrawn. Because claims 53-68 depend from, and, therefore, include all of the limitations of claim 52, it is respectfully submitted that these claims are also allowable for at least the reasons stated above.



RECEIVED  
CENTRAL FAX CENTER  
APR 02 2007

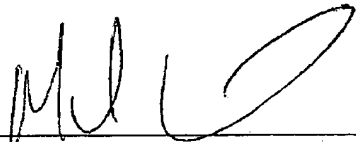
Attorney Docket No.: 40101/13901  
Ref. No.: 2008.005

CONCLUSION

It is therefore respectfully submitted that all of the presently pending claims are allowable. All issues raised by the Examiner having been addressed, an early and favorable action on the merits is earnestly solicited.

Respectfully submitted,

Dated: April 2, 2007

By:   
Michael J. Marcin (Reg. No. 48,198)

Fay Kaplun & Marcin, LLP  
150 Broadway, Suite 702  
New York, New York 10038  
Tel.: (212) 619-6000  
Fax: (212) 619-0276